

TARTU ÜLIKOOL  
MATEMAATIKA-INFORMAATIKA TEADUSKOND  
Arvutiteaduse Instituut  
Informaatika eriala

Margus Luik  
Robotmannekeeni käsitsijuhtimise tarkvara – Mannequin Manager  
Bakalaureusetöö (6 EAP)

Juhendaja: Anne Villems  
Kaasjuhendaja: Helina Kitsing

Autor: .....“ .....“ mai 2012  
Juhendaja: .....“ .....“ mai 2012

Lubada kaitsmisele  
Professor: .....“ .....“ .....“ 2012

# Sisukord

Sissejuhatus.....	3
1 Fits.me ja robotmannekeenid.....	5
1.1 Internetipoed.....	5
1.2 Robotmannekeen.....	7
2 Nõuded ja süsteemi üldine arhitektuur.....	10
2.1 Nõuded programmile.....	10
2.2 Nõuete põhjal koostatud kavand.....	11
2.3 Nõuetest ja kavandist tulenevad raskused.....	13
2.4 Keelest C# tulenevad raskused.....	13
2.5 Vahekihid Mannequin Manageri ja robotmannekeeni vahel.....	14
3 Mannequin Manageri loomisel kasutatud vahendid.....	15
4 Mannequin Manageri disain.....	17
4.1 3D mudel ja aktuaatoreid kujutlevad nooled.....	19
4.2 Aktuaatorite liigutamine 3D ruumi abil.....	20
4.3 Windows Exploreri jäljendamine küljemenüüs.....	22
4.4 Mannequin Manageri struktuur - mediaatori mall.....	22
4.5 Versioonihaldus ja Launcher.....	24
5 Arendusvõimalused.....	26
5.1 Unity GUIskin, kergesti muudetav välimus.....	26
5.2 Tahvelarvuti versioon.....	26
5.3 Mannequin Manager läbi internetilehitseja.....	27
5.4 Täpne kujumuutev mannekeeni mudel.....	27
Kokkuvõte.....	28
Summary.....	29
Viited.....	30
Lisad.....	32
Lisa 1. Mannequin Manageri ekraanipildid.....	32

## Sissejuhatus

Eestist pärit bränd Fits.me valiti maailmas tuntud äriuudiste kanali CNBC(*Consumer News and Business Channel*)<sup>[1]</sup> poolt aasta 2011 parima 20 läbilööva brändi<sup>[2]</sup> hulka. Fits.me tegeleb robotmannekeenide abiga rõivaste pildistamisega, et internetist rõivaid ostes saaksid ostjad näha, kuidas rõivaese istub nende kehakuju jälgendava roboti seljas. Kuna robotmannekeene on vaja liigutada inimeste kehadele vastavasse asendisse, siis on selle jaoks vaja ka vastavat tarkvara.

Käesoleva töö kirjeldab robotmannekeenide käsitsijuhtimise tarkvara Mannequin Manager disainimist ja loomist. Programmi loomisel oli vaja arvestada, et:

- uus programm suudaks täita kõiki olemasoleva robotmannekeeni juhtimistarkvara funktsioone,
- oleksid täidetud kliendi poolt programmile esitatavad nõuded ning
- oleksid täidetud TÜ Tehnoloogiainstituudi robotmannekeenide arendusmeeskonna poolt ette antud nõuded.

Olemasolev programm nõudis kasutajalt spetsiifilisi teadmisi robotmannekeenide ja programmi kohta, et arvuti külge ühendatud robotmannekeeni liigutada oleks võimalik. See väljendus programmi konfigureerimises, et programm roboti üles leiaks ja, et kasutajaliidese nupud vastaksid õigele robotmannekeeni versioonile. Võrreldes olemasoleva programmiga, pidi Mannequin Manager kasutaja asemel tegema ära kogu robotihalduse. Pärast roboti ühendamist arvuti külge saab kasutaja robotmannekeeni juhtida lihtsalt käivitades programmi Mannequin Manager. Kõik konfigureerimisega seotud tegevus toimub automaatselt taustal.

Suurt rõhku tuli pöörata ka Mannequin Manageri disainimise juures kasutajamugavusele ja programmi lihtsusele. Alustades programmi paigaldusest, kuni robotmannekeeni soovitud asendite tegemise ja salvestamiseni välja, pidi kõik olema tehtav ilma, et kasutaja oleks eelnevalt välja õpetatud või kasutajal tekiks vajadus lugeda juhendeid.

Töö on jagatud viieks peatükiks. Esimene annab ülevaate, mis on robotmannekeenid ja milleks neid kasutatakse. Teises peatükis on välja toodud tähtsamad Mannequin Managerile esitatud nõuded ning autori poolt välja pakutud üldine programmi disain. Kolmas peatükk tutvustab lühidalt autori poolt Mannequin Manageri valmistamiseks kasutatud töövahendeid. Neljas peatükk kirjeldab teises peatükis välja pakutud disaini põhjalikumalt ning selgitab

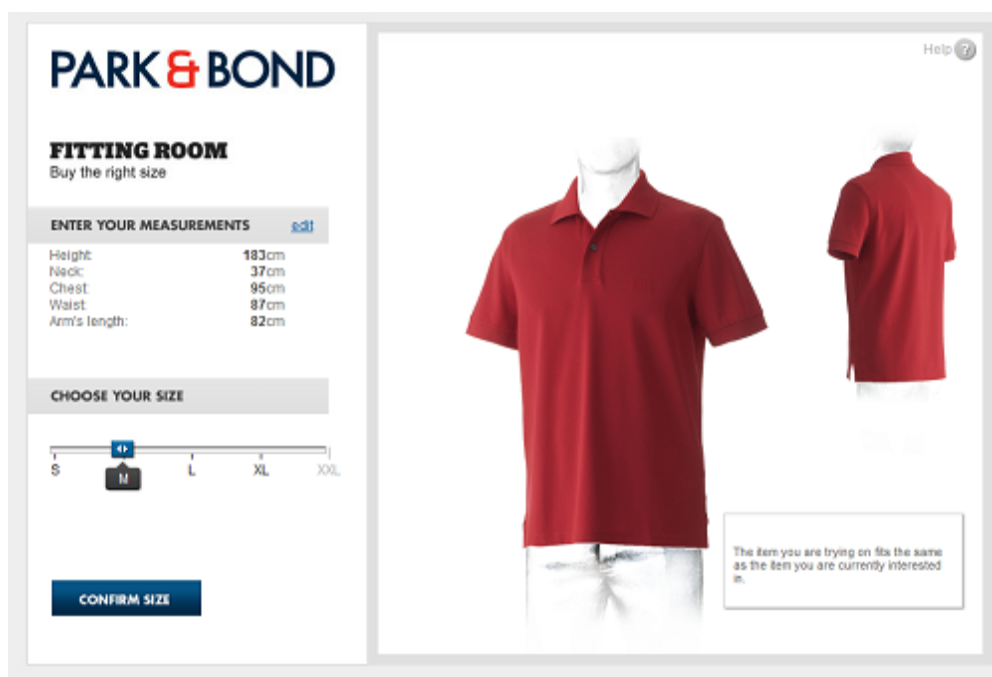
kuidas on antud disain implementeeritud. Töös on piirdutud on siiski vaid ülevaatega ning mõne algoritmi üldise kirjeldusega. Kuna programm on loodud töökeskkonna Unity 3D abiga, siis on hoidutud ka klasside skeemist, kuna selle eripärad nõuaksid Unity 3D süsteemi põhjalikumat tutvustust, mis juba ise-eneest ei ole mahu poolest võimalik sobitada ühte bakalaureusetöösse. Viies peatükk pakub välja võimalusi edasisteks Mannequin Manageri arendamise suundadeks. Lisas üks on ekraanipildid Mannequin Manageri funktsioonidest, mille kajastamiseks töös ei leitud ruumi.

# 1 Fits.me ja robotmannekeenid

Käesolevas peatükis on kirjeldatud robotmannekeeni juhtimistarkvara Mannequin Manager vajalikkust. Sellejaoks tutvustatakse firma Fits.me Holdings Ltd(edaspidi Fits.me) tegevust ning robotmannekeene.

## 1.1 Internetipoed

Rõivatootjatel, kes müüvad tooteid internetipoodides, on võimalik Fits.me-lt osta oma poe juurde virtuaalne proovikabiin. Selles saavad ostjad näha rõivaid ja hinnata nende sobivust kehakuju muutvate robotmannekeenide seljas (Joonis 1).



Joonis 1. Internetipoe proovikabiini näidis Fits.me koduleheküljelt. [3]

Virtuaalse proovikabiini tekitamiseks või garderoobi täiendamiseks peab rõivatootja uue riiete seeria valmimisel saatma igast mudelist kõigi suuruste näidised Fits.me fotostuudiosse. Studios (Joonis 2) pildistatakse rõivaid robotmannekeenide seljas erinevate kehakujude jaoks. Kuna kehakujusid on väga erinevaid, siis meeste jaoks juba 5 erineva riideeseme korral (suurustes XS, S, M, L, XL) tehakse studios kuni 700 pilti. Naiste rõivaste puhul on piltide arv kuni 5 korda suurem.



Joonis 2. Fotostuudio robotmannekeenide pildistamiseks.

Suure hulga piltide tegemiseks on pildistamise protsessi võimalikult palju automatiseeritud – automaatselt toimub mannekeenide asendite võtmine ja pildistamine. Käsitsi toimub mannekeeni riietamine.

Eeltööna on vaja teha mannekeenide automaatse pildistamise jaoks positsioonid. Nende tegemiseks on mõeldud programm Mannequin Manager, mille disainimisest ja programmeerimisest räägib käesolev töö.

## 1.2 Robotmannekeen

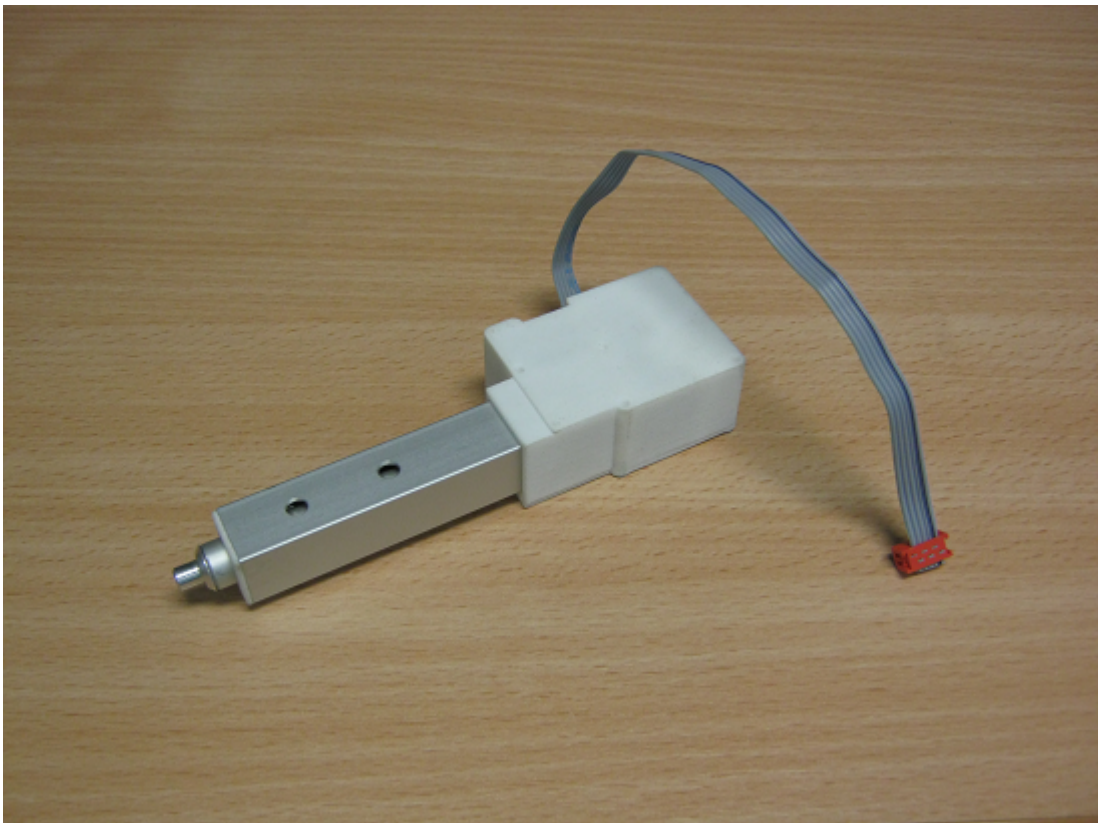
Robotmannekeenide ülesanne on jäljendada erinevaid inimeste kehakujusid teatud suurusvahemikes. Mannekeeni ehitus võimaldab muuta mõõte üle kogu keha, alates kaela ümbermõõdust kuni reite ümbermõõduni. Võimalik on muuta ka käte ümbermõõtu ning naismannekeenide puhul lisaks rindade suurust ja kõrgust. On oluline märkida, et sama ümbermõõt on saavutatav erinevate kehakujudega. Joonisel 3 on näha rõivasteta nais-robotmannekeeni kolmes erinevas asendis.



Joonis 3. Rõivasteta nais-robotmannekeen kolmes asendis.

Robotmannekeen muudab kehakuju aktuaatoritega. Meesmannekeenil on kokku 53 aktuaatorit, naismannekeenil 54. Definitsiooni <sup>[4]</sup> kohaselt on aktuaator iga objekt, mis muudab energiat mehaaniliseks liikumiseks. Selle definitsiooni kohaselt on ka iga mootor ja lihas aktuaator. Robootika seisukohalt on aktuaator seade, mis kindla sisendsignaali peale teeb soovitud mehaanilise liigutuse. Robotmannekeenil kasutatud aktuaatorid (Joonis 4) koosnevad:

- elektrimootorist,
- andurist, mis loeb liigutuse koordinaati,
- otsaandurist, mis paneb paika nullkoordinaadi,
- kruviülekandest, mis muudab pöörlemisliikumise sirgliikumiseks ning
- mikrokontrollerist.



Joonis 4. Robotmannekeenil kasutatav aktuaator.



Aktuaatorite rohkuse tõttu on vajalik lisamoodul, mis vahendaks teateid arvuti ja aktuaatorite vahel. Robotmannekeenil on selleks CAN-protokolli <sup>[5]</sup> realiseeriv plaat, lühidalt CAN-plaat.

CAN ehk *controller area network* on 1986 SAE International organisatsiooni poolt mootorsõidukite tarvis välja töötatud protokoll, mille alusel toimub seadmesisene informatsiooni liikumine. Näiteks autos peab mootorit juhtiv moodul teatama ülekannet juhtivale moodulile, kui mitu pööret sekundis teeb hetkel mootor. Selleks saadab mootorit juhtiv moodul välja teate CAN-ile ning CAN edastab selle teate kõigile teistele moodulitele. Olenevalt sõnumist otsustavad teised moodulid, mida nad selle informatsiooni kohaselt peaksid tegema.

CAN-i nõrkus on see, et korraga saab sõnumit saata ainult üks moodul. Sõidukites on aga tihti kuni 70 erinevat elektroonilist moodulit. Kui nendest kaks tükki üritavad korraga teadet saata, siis tekib põrge. Põrgete vältimiseks on igale moodulile määratud numbriline identifikaator. Kui kaks moodulit alustavad korraga andmete edastust, siis madalama identifikaatoriga moodul lõpetab teate saatmise ning ootab lühikese aja, et uuesti proovida. Ootamise ajal saab kõrgema identifikaatoriga moodul lõpetada oma teate edastuse. See tagab CAN-i töökindluse, et kõik teated saaksid edastatud ning tähtsamad teated jõuaksid alati enne läbi. Kui põrkeid on liiga palju, siis on ka võimalus, et teated hakkavad hilinebma kauem, kui nad olulised on. CAN protokollil on antud hinnang, et umbes 70%-80% andmeside läbilaskemahust võib täituda, enne kui teated hakkaksid hilinebma <sup>[5]</sup>.

Robotmannekeenid koosnevad rohkem kui 50 aktuaatorist, CAN plaadist, mis vahendab informatsiooni arvuti ning aktuaatorite vahel, sisekonstruktsioonist ning kattest. Peale elektrooniga ja mehaanika on roboti juures väga oluline ka tarkvara. Programmid jooksevad iga roboti aktuaatori ning ka CAN plaadi peal. Need kõik tegelevad teadete vastu võtmise, saatmise ning ka teadete põhjal käitumisega.

## 2 Nõuded ja süsteemi üldine arhitektuur

Käesolevas peatükis tutvustatakse autorile antud ülesannet ning autori poolt välja pakutud lahendust. Ülesanded autorile on esitatud programmi funktsionaalsete ning mitte-funktsionaalsete nõuetena.

### 2.1 Nõuded programmile

Käesoleva töö autori ülesandeks oli disainida ja programmeerida kasutajaliides, mis võimaldaks liigutada robotmannekeeni, oleks kerge kasutada ning näeks ka hea välja.

Lähtepunktiks programmi disainimisel ja loomisel olid nõuded, mis tulenesid olemasolevast mannekeeni juhtimise tarkvarast, Tartu Ülikooli tehnoloogiainstituudi robotmannekeeni arendusmeeskonnalt ja kliendi Fits.me poolt. Järgnevates nimekirjades väljatoodud funktsionaalsed ja mitte-funktsionaalsed nõuded on rühmitatud nende esitajate põhjal, mitte nende tüübi järgi.

Esiteks nimekiri nõuetest, mis olid seatud Mannequin Managerile olemasoleva juhtimisprogrammi poolt. Mannequin Manager peab suutma:

- liigutada mannekeeni kõiki aktuaatoreid soovitud suunas,
- salvestada mannekeeni asendeid csv (*comma sperated value*) tüüpi asendifailidesse,
- lugeda csv failist asendi ja liigutada mannekeeni sellesse asendisse,
- luua esitusloendit asenditest ja seda ette mängida,
- kuvada aktuaatorite olekuid ja veateateid.

Järgmisena nõuded Mannequin Managerile, mis tulid robotmannekeeni arendusmeeskonna poolt Tartu Ülikooli Tehnoloogiainstituudis. Mannequin Manager peab:

- olema kirjutatud keeles C#,
- töötama operatsioonisüsteemidel Windows 7 ja Windows Vista,
- suutma salvestada mitut asendit üheks esitusloendisse,
- olema lihtne ja mugav kasutada.

- olema kasutaja jaoks uuendatav üle interneti.

Viimasena nõuded, mis esitati programmile olemasoleva juhtimisprogrammi kasutajate poolt Fits.me stuudiotest. Mannequin Manager peab:

- tuvastama automaatselt arvutiga ühenduses oleva mannekeeni ja selle versiooni,
- võimaldama mannekeenide vahetust programmi töö ajal,
- mannekeeniga kaotatud ühendust taastada ilma kasutaja sekkumiseta,
- hoidma asendite faile eraldatult vastavalt mannekeeni versioonile,
- kuvama kasutajale ainult õige mannekeeni asendifailid.
- liigutama mannekeeni sümmeetriliselt vertikaalse kesktelje suhtes.

Kõikide nende nõuete põhjal sai koostatud programmi esialgne kavand. Järgmises punktis on kirjutatud, kuidas selleni jõuti ja mida see endast kujutas.

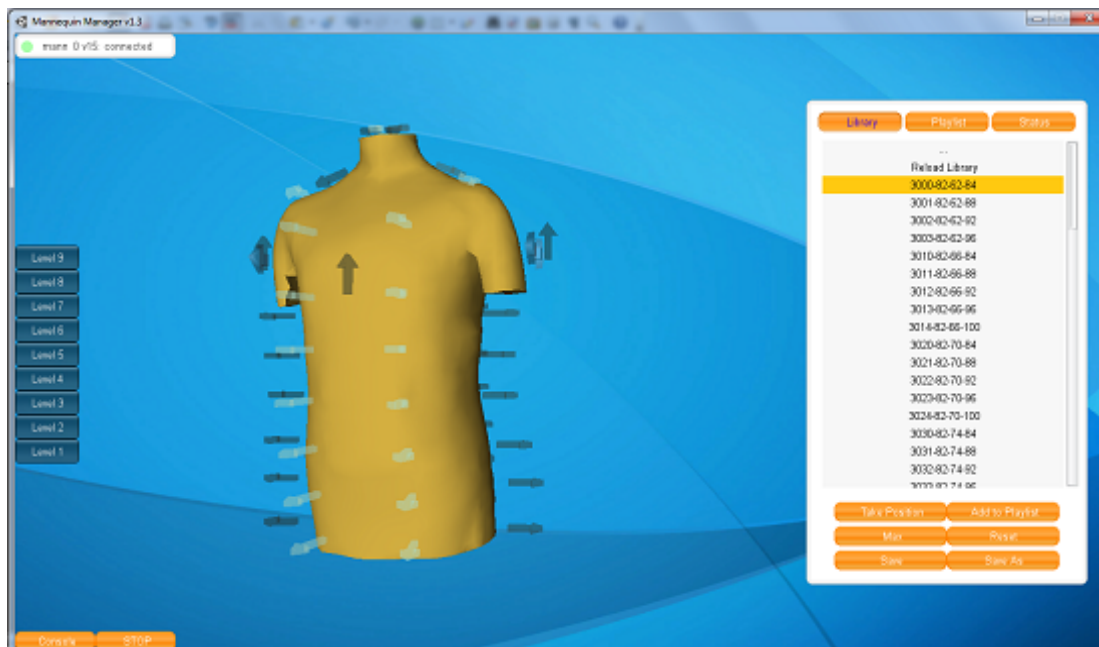
## **2.2 Nõuete põhjal koostatud kavand**

Kasutajaliidese peamine nõue on liigutada kõiki aktuaatoreid. See tähendab kuna aktuaatoreid on kuni 54 ja iga üks nendest vajab nuppe kahes suunas liikumiseks, siis kokku on vaja rohkem kui 108-t nuppu. Lisanduksid ka muud nupud asendite laadimiseks ja salvestamiseks, nimekirjad asendite haldamiseks, teadeteaken roboti seisukorra kuvamiseks ja mingil kujul informatsioon, mis paneb iga nupu vastavusse kindla aktuaatoriga. See tähendab rohkem nuppe ekraanil, kui neid on arvuti klaviatuuril ning nupud moodustavad vaid osa kõigest vajalikest. Ilmselgelt oleks sellist kasutajaliidest väga raske ja ebamugav kasutada.

Eelnev programm oli realiseeritud 11 erineva vaatega, mis grupeerisid aktuaatorite liigutamise nuppe korruste kaupa ning lisaks vaated failihaldusele ja robotilt tuleva tagasiside kohta. See muutis kõik piisavalt selgeks, et roboti juhtimine oleks võimalik pärast lühikest väljaõpet roboti sisestruktuurist, kuid üldine ülevaade robotist oli halb. Ka veateatest sai kasutaja teada alles siis, kui ta läks spetsiaalselt seda tagasisidevaatest otsima.

Probleemi lahendamiseks oli parim tee kolmemõõtmeline kasutajaliides (Joonis 5). See annab võimaluse saada ühte vaatesse korraga kõik nupud, paigutades need kolmemõõtmelise robotmannekeeni pinnale aktuaatoritega vastavuses olevatesse punktidesse.

Kasutajal on tänu antud disainile hea ülevaade kogu mannekeenist ning intuiitiivselt lihtne leida üles õige nupp soovitud aktuaatori liigutamiseks. Tulenevalt nuppude disainist, pakuvad need ka kasutajale tagasisidet.



Joonis 5. Kolme mõõtmelise kasutajaliidese realisatsioon.

Kolmemõõtmeline kasutajaliides hoiab ruumi kokku kahel põhjusel:

- Osa nuppe on ekraanilt peidetud, kuna need asuvad mannekeeni selja taga. Kuna mannekeeni pööramine on väga kiire tegevus, siis on ka mannekeeni taga peidus olevad nupud siiski vahetus kasutusläheduses.
- Nupp kolmemõõtmelises ruumis on vastavuses aktuaatoriga, mida see liigutab ning tänu sellele ei vaja selgitavat pilti aktuaatori asukoha jaoks.

Kolmemõõtmelise kasutajaliidesega programm pakkus suurepäraselt võimalust teha roboti liigutamine intuiitiivseks ning kasutajaliides kompaktses. Keeruliseks muutus aga programmi programmeerimine, kuna kolmemõõtmelise virtuaalmaailma loomine on keerulisem, kui kahemõõtmeliste programmide kirjutamine.

## **2.3 Nõuetest ja kavandist tulenevad raskused**

Lisaks 3D ruumile lisas programmile keerukust ka suhtlus robotmannekeeniga. Programmi jaoks tähendab suhtlus robotiga, et programm peab olema suuteline tegelema väga suure hulga ootamatute olukordadega, mida robot võib tekitada.

Kolmandaks raskuspunktiks sai TCP/IP ühenduse loomine ja hoidmine. Nimelt suhtleb kasutajaliides ühe vahekihina mannekeeni võrguühendusteenust pakkuva programmiga Daemon, mis Mannequinapi teegi abiga muudab TCP/IP pakette robotile arusaadavateks käskudeks.

Kokku tähendas see programmi loomist, mis on kolmemõõtmeline, suhtleb robotiga ning haldab võrguühendust. Kõik need kolm on omadused, mis võivad muuta programmi halva realisatsiooni korral ebastabiilseks.

## **2.4 Keelest C# tulenevad raskused**

C# puhul on tegemist imperatiivse objektorienteeritud keelega. See on välja töötatud ja arenduses Microsofti poolt. Keele eesmärk on olla lihtne, kuid sarnaneda koodis keeltele C ja C++. Võrreldes nimetatud keeltega, pakub C# programmeerimise lihtsustamiseks mälu haldamist.

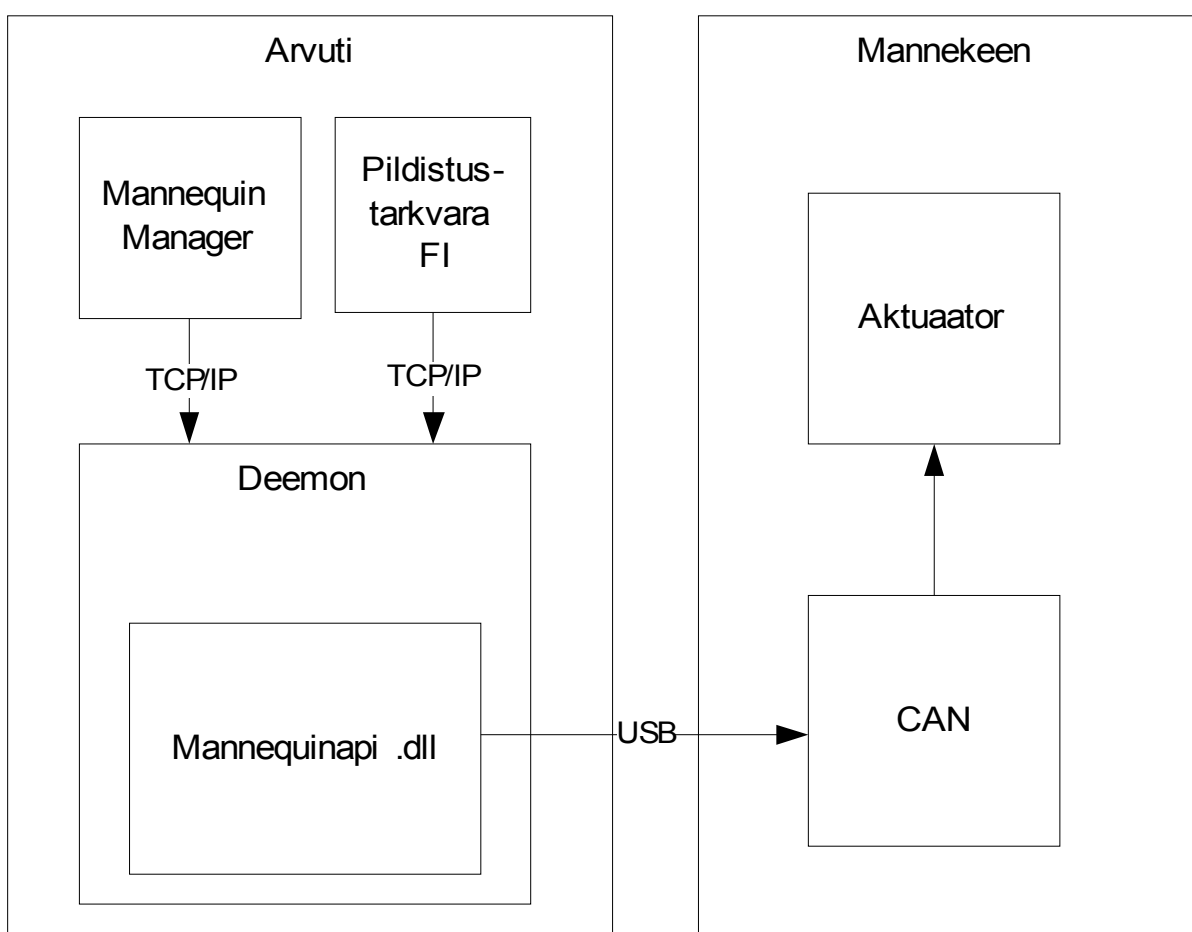
Kuigi keeles C# on kindlasti lihtsam programmeerida, kui keeles C++, tekitas keele C# valik teisi raskusi. Nimelt on C# ressursinõudlikum kui C++ ja sellepärast on reeglina kõik 3D teegid loodud kasutamiseks keelega C++. Pärast pikka C#-ga töötavate teekide otsingut jäi alles 2 mõistlikku valikut, kas kasutada programmi loomiseks mõnda OpenGL C# kohandust või kirjutada programm Unity 3D keskkonnas. Valik sai langetatud viimase kasuks, lootuses, et Unity 3D võimaldab luua programmi tunduvalt kiiremini ja lõpptulemus on 3D maailma loomisest tulenevate vigade suhtes stabiilsem.

Programmi kirjutamise käigus selgus, et otsene suhtlus robotiga käib üle jõu Unity 3D keskkonnale, mille peamine otstarve on 3D mängude ja internetirakenduste loomine. Üle jõu käimine väljendus programmi tööd lõpetavate vigadega Unity 3D mootoris. Kuna Unity 3D mootor on koodiosa, millele Unity keskkonnas programmi loojal ligipääs puudub, siis ainus väljapääs sellest olukorrast oli suhtlus robotiga suunata läbi programmi Daemon, mis on võrguteenindusprogramm Mannequinapi teegi jaoks.

Kokkuvõttes nõue lihtsamale keelele tähendas väga piiratud valikut 3D arendusvahendite vahel. Lisandus ka võrguühenduse programmeerimine, kuna Unity 3D ei ole suuteline stabiilselt koos töötama Mannequinapi teegiga. Kuidas täpselt andmed liiguvad Mannequin Manageri ja mannekeeni vahel, on selgitatud järgmises punktis.

## 2.5 Vahekihid Mannequin Manageri ja robotmannekeeni vahel

Mannequin Manageri ei suhtle robotmannekeeni aktuaatoritega otse, vaid saadab robotile mõeldud teated programmile Daemon. Joonisel 6 on näha, kuidas Mannequin Managerist antud liikumise käsk jõuab aktuaatorini.



Joonis 6. Andmeside Mannequin Manageri ja mannekeeni vahel.

Daemon on programm, mis võtab vastu läbi TCP/IP tulevaid käske ja kasutades Mannequinapi dünaamilist teeki annab selle teate edasi robotis asuvale CAN plaadile ning see omakorda aktuaatorile. Kuigi joonisel 5 on märgitud Daemon ja Mannequin Manager mõlemad ühte arvutisse, kuid TCP/IP kasutamisest tulenevalt võivad need olla ka erinevates arvutites.

### 3 Mannequin Manageri loomisel kasutatud vahendid

Mannequin Manageri loomisel kasutati mitmete olemasolevate töövahendite ja programmide abi. Järgnevalt on neid töövahendeid lühidalt tutvustatud.

- **Unity 3D** <sup>[6]</sup>

Tegemist on töökeskkonnaga, mis aitab luua kolmemõõtmelisi virtuaalseid maailmu. See koosneb wysiwyg („*What you see is what you get*“) 3D-redkatorist ja 3D objektide käitumise programmeerimiseks kasutatavast teegist nimega MonoScript <sup>[7]</sup>. MonoScripti on võimalik kasutada keeltega Boo, C# ja Javascript ning kood kompileeritakse Mono <sup>[8]</sup> (avatud lähtekoodiga platvormist sõltumatu versioon .Net Frameworkist) abiga CLI baitkoodiks <sup>[9]</sup>. Unity 3D töökeskkonna ja MonoScripti tutvustamine ei ole antud töö puhul kahjuks mahuliselt võimalik.

- **3D skänner**

Mannequin Manageris kasutatava 3D mannekeenide mudelid on sisse skännitud Tartu Ülikooli Tehnoloogiainstituudis asuva 3D täiskere laserskänneriga. Tegemist on seadmega, mis teeb inimesest või esemest täpse koopia arvutisse.

- **Geomagic Studio** <sup>[10]</sup>

Tegemist on 3D skännide puhastamiseks ja töötlemiseks ette nähtud tarkvaraga. 3D skännerite puhul on tegemist väga uue tehnoloogiaga. Esimene algeline 3D skänner tehti aastal 1996 <sup>[11]</sup>. Sellest ajast on küll tehnoloogia väga palju edasi arenenud, kuid siiski tekitavad ka praegused skännerid suurel hulgal müra skännile. See väljendub üleliigsete punktide ning karedate pindadena.

Geomagic Studio pakub mugavat võimalust käsitsi eemaldada soovimatuid punkte skännist ning ka mitmesuguseid algoritme skänni pindade silumiseks, üleliigsete punktide otsimiseks ning punktidest pindade loomist. Selle programmiga loodi skännist korralik mannekeeni kujutis Mannequin Manageri jaoks.

- **Blender** <sup>[12]</sup>

Blender on vabavaraline 3D kujutiste loomise programm. Blenderit kasutati Mannequin Manageri loomisel ainult aktuaatoreid kujutavate noolte modelleerimiseks.

- **Gimp** <sup>[13]</sup>

Gimp on vabavaraline piltide manipuleerimise ja joonistamise programm. Mannequin Manager kasutab suurel hulgal Unity 3D-ga kaasnevat 2D graafikat menüüde kujundamisel, kuid mõningad ikoonid on loodud ka käesoleva töö autori poolt kasutades Gimp-i.

- **7-Zip** <sup>[14]</sup>

7-Zip on vabavaraline arhiveerimise programm. See võimaldab ühe või mitu faili pakkida kokku üheks 7z laiendiga failiks, mis võtab vähem ruumi, kui esialgsed failid kokku. Need arhiivid on kasutuses Mannequin Manageri uuenduste kohale toimetamiseks. Uuendused kokkupakitult 7-zip-ga on umbes 30% väiksemad. Sellest on palju kasu, kuna uuendused toimetatakse kohale üle interneti.

- **7-Zip SFX Maker** <sup>[15]</sup>

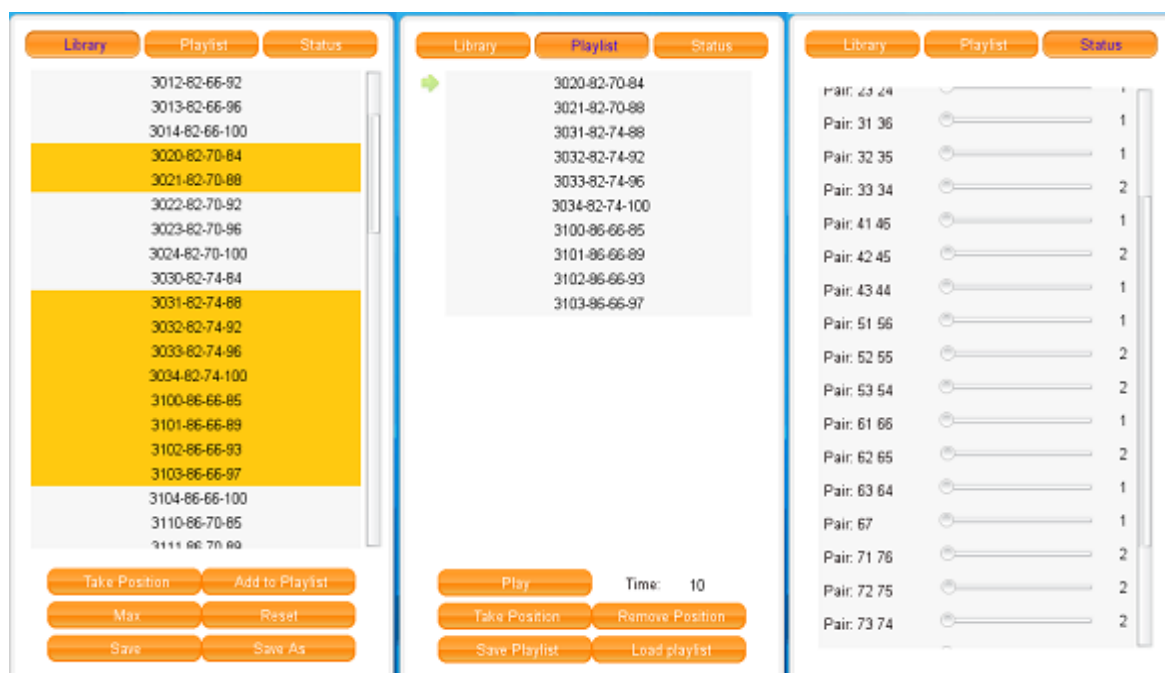
7-Zip SFX Maker on vabavaraline programm, mis muudab 7z laiendiga faile ennast ise lahti pakkivateks arhiivideks. Need on kasutuses Mannequin Manageri uuendusprotsessis.



## 4 Mannequin Manageri disain

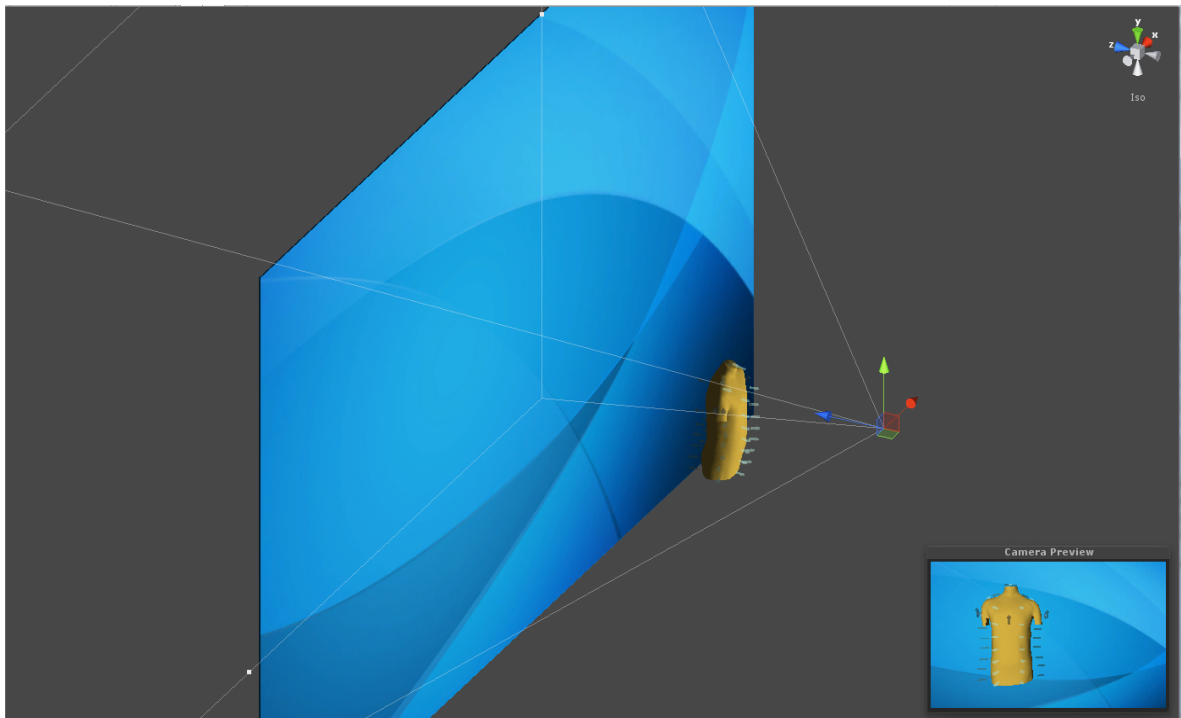
Mannequin Manager on disainitud keskse kolmemõõtmelise mannekeeni ümber, mis võimaldab mannekeeni liigutamist ja annab tagasisidet aktuaatorite kohta. Mudelile lisaks on kasutajaliidese paremal küljel küljemenüü, kuhu on pandud kogu funktsionaalsus, millele ligipääs kolmemõõtmelisest ruumist ei pakuks mingit eelist. Need funktsioonid on jagatud kolme vaatesse – *library*, *playlist* ja *status*. Küljemenüüd on kujutatud joonisel 7.

- Library: Tegeleb asendite salvestamisega ja laadimisega failidesse, võimaldab lisada asendeid esitusloendisse ning saata mannekeenile taaskäivitus (*restart*) käsklust.
- Playlist: Tegeleb esitusloendite ettemängimise, salvestamise ning laadimisega. Võimalik on valida ka esitusloendi ettemängimise kiirust.
- Status: Kuvab korraga kõikide aktuaatorite koordinaadid identifikaatorite põhjal ning võimaldab aktuaatorite liigutamist kindlale klaviatuurilt sisestatud koordinaadile.



Joonis 7. Mannequin Manageri küljemenüüd.

Järgnevalt tutvustatakse, mis on 2D ja 3D elemendid või objektid programmis. 2D elementideks loetakse 3D programmide puhul kõike, mis eksisteerib ainult ekraani pinnal. Võrdluseks, 3D elemendid eksisteerivad kujutletavas ruumis ning ekraanil on näha nende projektsioon. Joonisel 8 on näha 3D objekte ja nende projektsiooni ekraanile. Joonisel on kujutatud mannekeen, pind tausta jaoks ning virtuaalne kaamera. Joonise alumises paremas servas on näha ruumi projektsioon kaamera vaatepunktist.



Joonis 8. Mannequin Manageri 3D objektid ja projektsioon kaamera vaatepunktist.

Joonisel 9 on kujutatud ainult Mannequin Manageri 2D elemendid. Nendest peamised on nupud, kastid ja kirjad. Kastid võivad sisaldada ka pilte, tänu millele on võimalik kujundada just selline kasutajaliides, nagu soov on.



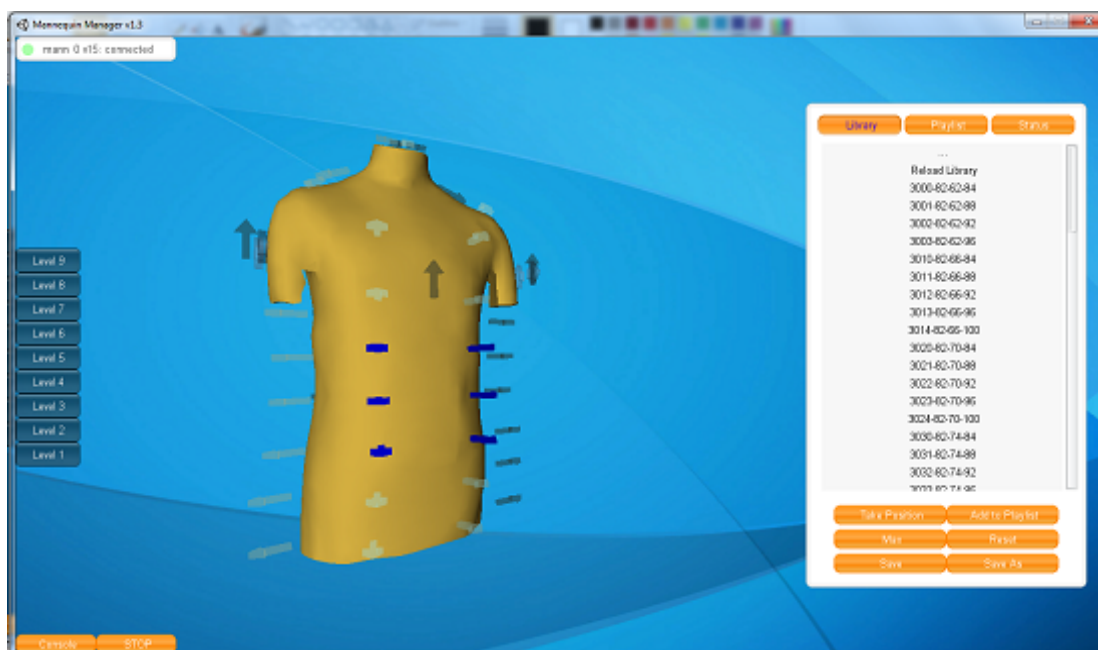
Joonis 9. Mannequin Manageri 2D elemendid.

Käesoleva peatüki järgnevad alampunktid tutvustavad Mannequin Manageri 3D objekte ja 2D elemente lähemalt, alustades 3D mudelist.

#### 4.1 3D mudel ja aktuaatoreid kujutlevad nooled

Mannequin Manager kuvab kesksel kohal 3D mudelit mannekeenist. Tegemist on pööratava, aga mitte kuju muutva mudeliga. Aktuaatorite ja mannekeeni keha kuju muutmise kujutamiseks on mudeli juures hõljuvad nooled (Joonis 10). Iga nool vastab ühele aktuaatorile ning muudab suurust vastavalt aktuaatori asendile. Igal noolel tekib ka hiirega üle liikudes vastav teade, mis kuvab aktuaatori asendit ja olekut. Aktuaatori olekud ja nendele vastavad värvid on järgmised.

- Helesinine transparentne nool on töövalmis valimata aktuaator.
- Tumesinine nool on töövalmis valitud aktuaator.
- Kollane nool on hetkel liikuv aktuaator.
- Punane nool on aktuaator, mille töös on tekkinud viga.



Joonis 10. Mannekeeni mudel koos töövalmis aktuaatoreid kujutavate nooltega.

Nendest olekutest kaks esimest omavad sisulist erinevust ainult Mannequin Manageri raames ning mannekeeni poolest on tegemist sama olekuga. Järgnev punkt kirjeldab, mida need olekud tähendavad ning kuidas kasutaja saab neid olekuid vahetada.

## 4.2 Aktuaatorite liigutamine 3D ruumi abil

Eelnevas punktis mainiti valimata ja valitud noolte olekut. Kuna programmis aktuaatoreid kujutavate noolte puhul ei ole tegemist otseselt nuppudega, vaid täielikult käesoleva töö autori poolt loodud kontrollelementidega, siis oli võimalik disainida aktuaatorite liikumiseks midagi paremat, kui lihtsalt suuremaks või väiksemaks nupp.

Liigutamise süsteem koosneb kolmest osast.

- 3D objektide loomise programmis Blender loodud 3D noolte kujundid.
- Igale noolele vastav aktuaatorite andmeid haldav skript Attributes.
- 2D ekraanilt hiireklõpsude põhjal 3D ruumist nooli valiv skript MouseSelect.

Iga noolega kaasas käiva Attributes skripti ülesanded on täpsemalt:

- hoida noolega seotud aktuaatori olekut,
- hoida noolega seotud aktuaatori koordinaati,

- muuta noole suurust vastavalt aktuaatori koordinaadile,
- muuta noole värvi vastavalt olekule,
- kuvada ekraanile vastava aktuaatori informatsiooniga, kui hiire kursor on noole kohal,
- muuta aktuaatori koordinaati, kui nool on valitud olekus ja kasutaja liigutab hiire rulli või vajutab klaviatuuril üles/alla klahvi,
- informeerida aktuaatori koordinaadi muutmisest moodulit, mis saadab robotile liikumise käske.

MouseSelect skripti ainus ülesanne on muuta noolte olekuid, kas valitud või valimata seisundisse. Vaatame valitud ja valimata probleemi lähemalt.

Üks võimalus kasutajal nooli valida on otse noole peale vasaku hiirenupuga vajutades. Sellisel juhul tekitab MouseSelect skript virtuaalse kaamera vaatest kiire 3D ruumi. Juhul, kui see kiir põrkub mõne aktuaatoriga, siis saadab MouseSelect skript selle noole Attributes skriptile teate muuta nool valitud olekusse. Juhul, kui antud kiir tabab mannekeeni või ei taba aktuaatorit, siis kontrollib skript, kas klaviatuuril on hetkel alla vajutatud shift klahv. Kui shift klahv all ei ole, siis saadab MouseSelect teate kõigile Attributes skriptidele minna valimata olekusse.

Teine võimalus kasutajal nooli valida on sarnaselt Windowsis kasutatavale ristküliku vedamisele. Selle jaoks joonistab MouseScript temaga seotud kaamera vaatesse ristküliku, kui kasutaja vajutab alla hiire parema klahvi. Hetkel, kui kasutaja laseb hiire paremast klahvist lahti, siis MouseSkript kontrollib ristküliku sisse jäänud kaamera projektsioonis asuvaid objekte ning kui tegemist on mõne noolega, siis saadab nende noolte Attributes skriptile teate lülitada valitud olekusse. Kõik ülejäänud Attributes skriptid saavad teate minna valimata olekusse. Ka siin võetakse arvesse klaviatuuri shift klahvi.

Esimese ja teise valmisvõimaluse oluline erinevus on, et esimesel juhul tehakse kontroll 3D ruumis noolte jaoks, kuid teisel juhul kontrollitakse 3D ruumi kaamera 2D projektsioonis hiirega tekitatud ristküliku sisse jäänud noolte koordinaate. See tähendab, et ristkülikuga valimise puhul peab ristküliku sisse jääma noole keskpunkt, kuid esimesel juhul töötab valik noole suvalise punkti puhul.

### 4.3 Windows Exploreri jäljendamine küljemenüüs

Sarnaselt noolte valikuga on Windowsi stiilis shift ja ctrl klahvi kasutust emuleeritud ka küljemenüüs. Sellise käitumise eesmärk on hõlbustada korraga mitme asendi lisamist esitlusloendisse.

Hoides all shift klahvi, siis valitakse kõik asendid eelmise ja praeguse valitud asendi vahel nimekirjas. Kõik ülejäänud asendid muudetakse mitte valituks. Kui ühtegi asendit ei ole valitud, siis shift klahv valib kõik asendid esimesest kuni asendini, mille peale vajutati.

Hoides all ctrl klahvi, jääb uue asendi valimisel eelmiste valitud asendite mitte valituks muutmine ära. Kaetud on ka kõik erandlikud olukorrad ning shift ja ctrl klahvi koos kasutamine.

Siiani on kirjeldatud disaini kasutajat puudutavast vaatepunktist, järgmine alampeatükk räägib mallidest ja disainist programmi struktuuri mõttes.

### 4.4 Mannequin Manageri struktuur - mediaatori mall

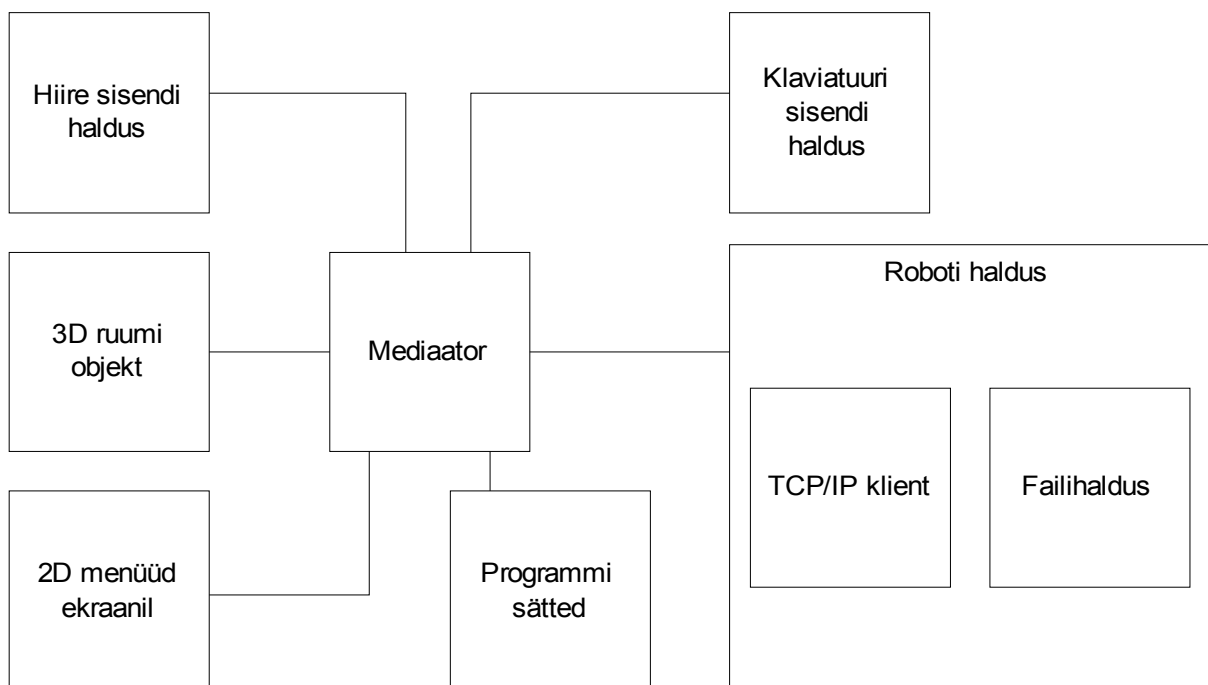
Alustuseks on vajalik teada, mida kujutab endast mall. Järgnev lühikirjeldus malli <sup>[16]</sup> kohta on kirjutatud wikipedia põhjal. Mall on programmeerimises taaskasutatav lähenemine teatud probleemi tarvis. Reeglina on mall väga üldine ega ole valmislahendus, mida saaks kasutada koodis. Mall ei sõltu otseselt ka kasutatavast programmeerimiskeelest, erandina, et keel peab olema objektorienteeritud.

Mediaatori mall <sup>[17]</sup> on disain, mis laseb objektidel olla teine-teisest võimalikult sõltumatud. See on teostatud keskse vahendava objektiga, mida nimetatakse mediaatoriks. Kõik ülejäänud objektid suhtlevad ainult mediaatorobjektiga ja otsene omavaheline suhtlus ei ole lubatud. Erandiks on objektide alamobjektid, millega suhtleb iga objekt ise. Mediaatorobjekti ja kõiki objekte, mis suhtlevad otse mediaatoriga nimetatakse programmi mooduliteks. Selline modulaarne disain aitab kergesti programmi lisada täiesti uusi objekte või vanu eemaldada, kuna iga mooduli lisamisel peab ainult kirjeldama suhtluse uue mooduli ja mediaatori vahel.

Siinkohal toaksin elulise näite, kuidas mediaator mall töötab väljastpoolt programmeerimist. Koht, kus sarnane suhtlus on igapäevaselt kasutuses, on lennujaam. Kui lennuk läheneb lennujaamale ja soovib maanduda, siis on oluline, et korraga maanduks ühel rajal ainult üks lennuk. Kui ühel ajal üritaks maanduda kaks või enam lennukit samal rajal,

siis oleks tõenäoliselt tulemus katastroofiline. Niisiis on igale lennukile oluline, et teine lennuk samal ajal ei maanduks samal rajal. Üks moodus selle saavutamiseks oleks, kui piloot võtaks ühendust kõigi teiste lennukitega ja lepiks kokku, kes asub nüüd maanduma. Kuna suuremates lennujaamades soovib korraga maanduda kümneid lennukeid, siis on kerge aru saada, et selline lähenemine ei ole eriti efektiivne. Veelgi enam, kui ühendust ei õnnestu võtta kõikide pilootidega, siis on tulemus ettearvamatu. Selle vältimiseks on igas lennujaamas hoopis dispetšer, kes juhendab kõiki piloote, millal ja millisele rajale nad peavad maanduma. Selles näites täidab dispetšer mediaatori ülesannet - kõik piloodid peavad suhtlema ainult ühe torniga ning torn juhendab kõiki piloote.

Järgnev joonis 11 illustreerib mediaatori malli kasutamist Mannequin Manageris.



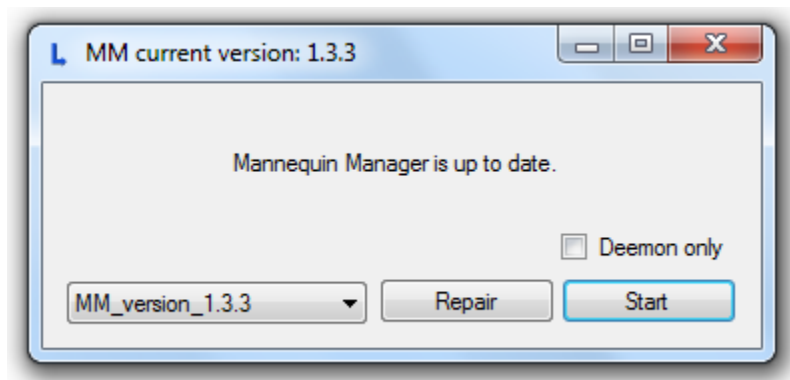
Joonis 11. Mediaatori mall Mannequin Manageris.

Joonisel 11 kujutatud Mannequin Manageri disainis on lubatud mitu eksemplari moodulitest 3D ruumi objekt ja roboti haldus. 3D ruumi objektide puhul moodustab ühe mooduli iga aktuaator ning roboti halduse moodul luuakse iga ühenduses oleva mannekeeni jaoks oma.

Mediaatori mall vähendab ohtu, et Mannequin Manageri ühes moodulis tehtud uuendused tekitaks vigu mõnes teises moodulis. Järgmine peatükk tutvustab, kuidas tehtud uuendused jõuavad Mannequin Manager kasutajateni.

## 4.5 Versioonihaldus ja Launcher

Launcher (Joonis 12) on programm, mille esialgne eesmärk oli Mannequin Manageri uuendamine. Kuid kuna Mannequin Manager on väga tihedalt seotud Deemoni ja Mannequinapi-ga, siis uuendab Launcher neid programmi ühtse paketina.



Joonis 12. Mannequin Manageri, Deemoni ja Mannequinapi uuendamise programm.

Launcher on pigem versioonihaldus programm, kui lihtsalt uuendus, kuna see võimaldab kasutajal väga lihtsalt mõnele eelmisele programmiversioonile tagasi minna.

Selline lahendus sai loodud, kuna Fits.me fotostudio, kus tarkvara on kasutusel, töötab väga tiheda graafikuga mitmes vahetuses. Juhuks, kui uus versioon programmist peaks sisaldama mõnda viga, siis ei peata see studio tööd, kuna vanale versioonile tagasi vahetamine 10Mb-se internetist allalaadimise kiiruse juurse võtab aega alla minuti.

Programmide uuendamise juures on peamine raskuspunkt see, et uuendatav programm ei tohi olla hetkel töös. Kuna oli vajalik ka Launcheri enda uuendamine, siis on uuendused pakitud arhiividesse ise lahti pakkivateks arhiivideks. Selleks on kasutades programmi 7-zip, mille väljundiks on .7z arhiivid. Need arhiivid muudetakse omakorda ise lahti pakkivateks arhiivideks, ehk SFX failideks, kasutades programmi 7-zip SFX Maker.

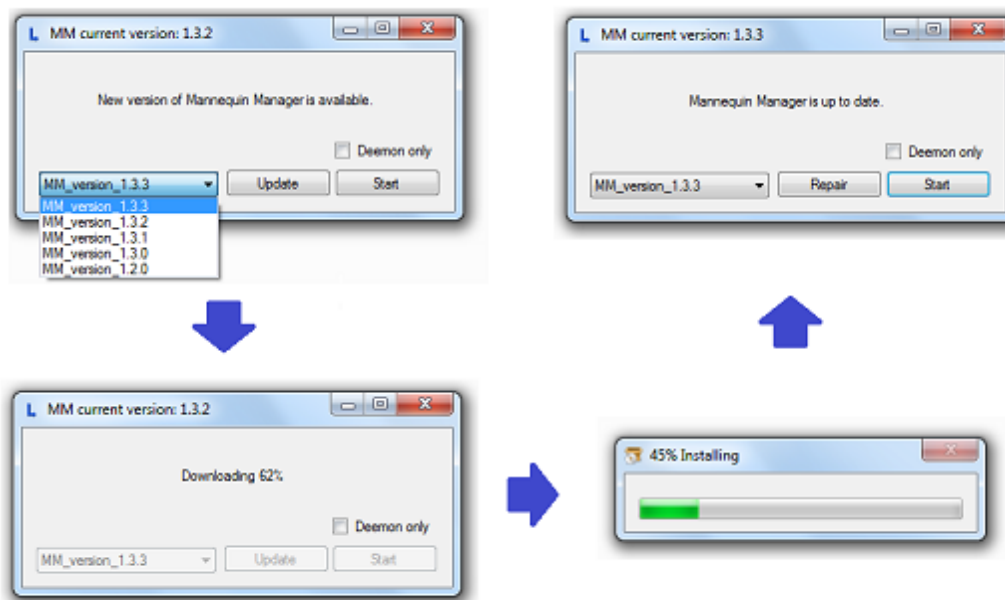
Uuenduse paigaldus toimub järgnevalt.

1. Launcher küsib Tartu Ülikooli Tehnoloogiainstituudi serverilt saadaval olevate versioonide nimekirja.
2. Kasutaja valib välja soovitud versiooni.
3. Laetakse alla valitud versioonile vastav SFX fail. SFX failide suurus on umbes 6 megabaiti.



4. Launcher kontrollib, kas Daemon või Mannequin Manager töötavad ning vajadusel sulgeb need.
5. Launcher käivitab Windowsi käsureaprogrammi cmd.exe käsuga oodata 3 sekundit ning seejärel käivitada uuendust sisaldav SFX fail.
6. SFX fail pakib lahti uuenduse.
7. SFX fail käivitab Launcheri.
8. SFX fail kustutab end.

Kogu kirjeldatud uuendamise protsess toimub kasutaja jaoks ühe nupuvajutusega. Joonisel 13 on kujutatud uuendamise protsess kasutaja vaatepunktist.



Joonis 13. Uuendamise protsess kasutaja vaatepunktist.

Käesolevas peatükis kirjeldati, millised valikuid tehti Mannequin Manageri disainis. Valiku tegemine disainis tähendab, et mõned lisad programmile on võimalikud ning teised ei ole. Järgmine peatükk räägib Mannequin Manageri arendamise võimalustest, mis ei eelda väga suuri muudatusi käesolevas töös kirjeldatud disainis.

## **5 Arendusvõimalused**

Käesolevas peatükis on kirjeldatud erinevaid ideid Mannequin Manageri edasiarenduseks. Siin olevad ideed on välja toodud, kuna need ei nõua suurt hulka muutusi programmi disainis.

### **5.1 Unity GUIskin, kergesti muudetav välimus**

Mannequin Manager kasutab Unity 3D-le sisseehitatud funktsionaalsust 2D kujunduse haldamiseks. See seob soovitud pildid ja tekstistiilid üheks grupiks, mille nimetus on GUIskin ning igale pildile märgitakse stiili tüüp. Kui kõik nupud ja kastid omavad samuti märgistatud stiili tüüpi, siis piisab vaid GUIskin-i vahetamisest, et muuta kogu kasutajaliidese välimust.

### **5.2 Tahvelarvuti versioon**

Ühe võimaliku arendusena võiks Mannequin Manager töötada ka tahvelarvutitel. Unity 3D toetab tarkvara kohandamist paljudele erinevatele platvormidele. Nende hulgas ka Android ja iOS. Kuigi ümberkompileerimine oleks väga lihtne, käib kohandamisega kaasas ka rida muid muutusi. Peamine osa, mis vajaks muutmist, on hiire ja klaviatuuri sisendid, kuna tahvelarvutitel puuduvad mõlemad. Nende asemel on sisendiks puuteekraan. Kuigi ekraanile on võimalik kuvada klaviatuuri, siis see võtaks ära liialt palju ruumi ülejäänud kasutusliidese jaoks.

Tahvelarvuti versiooni puhul saaks ka ära kasutada TCP/IP ühendust Mannequin Manageri-i ja Deemoni vahel. Nii on võimalik mannekeenrobot siiski ühendada arvuti külge ja sinna peale panna jooksmas Deemon ning Mannequin Manager saaks töötada tahvelarvuti peal, mis on ühenduses üle juhtmevaba internetivõrgu Deemoniga.

Tahvelarvuti peal olev kasutusliides oleks kindlasti mugavam kasutada asendite tegemise juures, kuna suur osa ajast kulub roboti mõõdistamisele. Nii saab asendeid valmistades olla stuudio töötajal tahvelarvuti ühes käes ning mõõdulint teises, mis kaotaks ära pideva vajaduse liikuda arvuti ja mannekeeni vahet. Tegemist oleks ka väga efektse mannekeeni juhtimisega demonstratsioonide raames.

### **5.3 Mannequin Manager läbi internetilehitseja**

Erinevalt tahvelarvuti versioonist, saaks internetilehitsejas töötav Mannequin Manager kasutada klaviatuuri ja hiirt, kuid muutmist vajaks hoopis failide haldus, kuna see peaks jääma stuudio, mitte kasutaja poolele.

Ainus kujutatav kasutus sellisele Mannequin Managerile-le oleks puhtalt reklaami mõttes. Kasutajad saaksid kordamööda lühikeseks ajaks võtta kontrolli roboti üle, seda liigutada ning veebikaamerast toimuvat jälgida.

### **5.4 Täpne kujumuutev mannekeeni mudel**

Tartu Ülikooli Tehnoloogiainstituudis arendatakse matemaatilist mannekeeni mudelit. Selle mudeli eesmärk on skännitud inimesele panna vastavusse robotmannekeeni asend.

Tulevikus on võimalik see mudel integreerida Mannequin Manageri, mis võimaldaks otse kasutajaliidesest näha, milline kuju on robotil ja virtuaalse mudeli põhjal teha mõõtmisi. See aitaks teha mannekeenile asendeid palju kiiremini ning ka täpsemini, kuna praegu on võimalik asendite suurust mõõta mannekeeni pealt või skännitud kujude põhjal paika seada. Nendest kahest esimene ei ole väga täpne ning teine on väga aeglane. 3D kujumuutev mudel annaks suurema kiiruse kui käsitsi mõõtmine ja sama täpsuse, mida suudab pakkuda 3D kehaskänner.

## Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli disainida ja programmeerida kolmemõõtmeline kasutajaliides, nimega Mannequin Manager, robotmannekeenide juhtimiseks.

Mannequin Managerile esitatud nõuetest üks olulisemaid oli, et programm peab olema lihtsasti kasutatav ka ilma väljaõppeta inimestele, kes ei oma infotehnoloogia alast haridust või tausta. Praegu on Mannequin Manager kasutuses Fits.me fotostuudiotest fotograafide poolt, kelle tagasiside põhjal on programmi kasutamine lihtne.

Mannequin Manageri on kasutatud ka koolinoorte poolt saate Rakett69<sup>[18]</sup> käigus. Võistlejatele anti ülesanne mõõta modelle ning seadistada robotmannekeene modellide kehakujudele vastavatesse asenditesse. Mõlemad meeskonnad kasutasid ülesande lahendamiseks Mannequin Manageri ja said ilma ettevalmistava koolitusest hakkama.

Antud tööst sai autor palju kogemusi tarkvara projekti läbiviimises ja 3D virtuaalmaailma loomises. Virtuaalmaailmaga seotud teadmised käivad peamiselt Unity 3D kohta. Sellest hoolimata jääb autoril tõenäoliselt viimaseks korraks kasutada Unity 3D süsteemi. Selle töövahendi eesmärk oli projekti lihtsustada, kuid töö käigus tekitas see mitmeid uusi probleeme, mille lahendamiseks tuli esialgset disaini muuta. Arvatavasti on siiski võimalik neid kogemusi üle kanda mõne teise sarnase süsteemi kasutamiseks või OpenGL ja Direct3D teekidega otse 3D graafika programmeerimisele.

## **Summary**

Mannequin Manager – Manual Control Interface for Robotic Mannequin

Bachelor Thesis

Margus Luik

## **Summary**

The aim of this thesis was to create a manual control interface, named Mannequin Manager, for shape-shifting robotic mannequins. The thesis discusses the requirements set for the program, the design solution proposed by author and the implementation of set design.

The thesis is composed of five parts. The first part of thesis explains what are robotic mannequins and how they are used in business practices. The second part gives an overview of requirements for program and introduces proposed design. Third part introduces tools used for programming and creation of assets. In the fourth part an overview of implementation is given to the design proposed in the second part. Fifth part proposes ideas for further development of Mannequin Manager.

As a result of this bachelor thesis a program was created and is now used by photographers in Fits.me photoshoot studios. It was also used by school youth in an estonian educational TV show Rakett69, where they had to adjust robotic mannequins body shape to the same of models. Both contesting teams completed the task to maximum amount of points, using Mannequin Manager without being previously trained.

## Viited

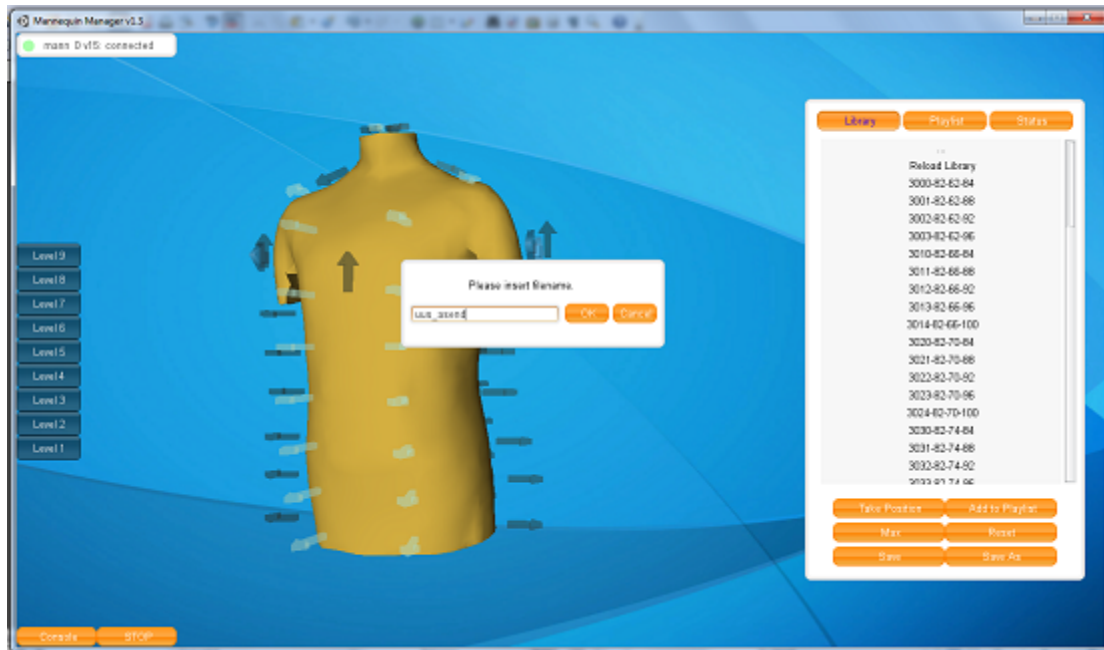
1. CNBC ametlik lehekülg [Võrgumaterjal][Viimati alla laetud 10.05.2012]  
<http://www.cnn.com/>
2. CNBC artikkel *Stand By For Action*[Võrgumaterjal][Viimati alla laetud 10.05.2012]  
<http://www.cnnmagazine.com/story/stand-by-for-action/1273/2/>
3. Fits.me ametlik lehekülg [Võrgumaterjal][Viimati alla laetud 10.05.2012]  
<http://fits.me/>
4. *Wisegeek, What is an Actuator* [Võrgumaterjal][Viimati alla laetud 10.05.2012]  
<http://www.wisegeek.com/what-is-an-actuator.htm>
5. *Wikipedia CAN protokoll* [Võrgumaterjal][Viimati alla laetud 10.05.2012]  
[http://en.wikipedia.org/wiki/CAN\\_bus](http://en.wikipedia.org/wiki/CAN_bus)
6. Tarkvara Unity 3D ametlik lehekülg [Võrgumaterjal][Viimati alla laetud 10.05.2012]  
<http://unity3d.com/>
7. Unity 3D *scripting reference*[Võrgumaterjal][Viimati alla laetud 10.05.2012]  
<http://unity3d.com/support/documentation/ScriptReference/MonoScript.html>
8. Mono ametlik lehekülg [Võrgumaterjal][Viimati alla laetud 10.05.2012]  
[http://www.mono-project.com/Main\\_Page](http://www.mono-project.com/Main_Page)
9. CLI baitkoodi tutvustus [Võrgumaterjal][Viimati alla laetud 10.05.2012]  
[http://www.mono-project.com/Introduction\\_to\\_developing\\_with\\_Mono](http://www.mono-project.com/Introduction_to_developing_with_Mono)
10. Tarkvara *Geomagic Studio* ametlik lehekülg[Võrgumaterjal][Viimati alla laetud 10.05.2012] <http://www.geomagic.com/>
11. Jerry Isdale, *A Bried History of 3D scanning*[Võrgumaterjal][Viimati alla laetud 10.05.2012] [http://vr.isdale.com/3DScanners/3d\\_scan\\_history/history.htm](http://vr.isdale.com/3DScanners/3d_scan_history/history.htm)
12. Tarkvara Blender ametlik lehekülg [Võrgumaterjal][Viimati alla laetud 10.05.2012]  
<http://www.blender.org/>
13. Tarkvara GIMP ametlik lehekülg [Võrgumaterjal][Viimati alla laetud 10.05.2012]  
<http://www.gimp.org/>
14. Tarkvara 7-Zip ametlik lehekülg [Võrgumaterjal][Viimati alla laetud 10.05.2012]  
<http://www.7-zip.org/>
15. Tarkvara 7-Zip SFX maker ametlik lehekülg [Võrgumaterjal][Viimati alla laetud 10.05.2012] <http://sfx-maker.sourceforge.net/>

16. Wikipedia, *Software design pattern* [Võrgumaterjal][Viimati alla laetud 10.05.2012]  
[http://en.wikipedia.org/wiki/Software\\_design\\_pattern](http://en.wikipedia.org/wiki/Software_design_pattern)
17. Source Making, *Mediator Design Pattern*[Võrgumaterjal][Viimati alla laetud 10.05.2012] [http://sourcemaking.com/design\\_patterns/mediator](http://sourcemaking.com/design_patterns/mediator)
18. ETV arhiiv Rakett69, 24.03.2012 19:35 [Võrgumaterjal][Viimati alla laetud 10.05.2012] <http://etv.err.ee/arhiiv.php?id=126298>

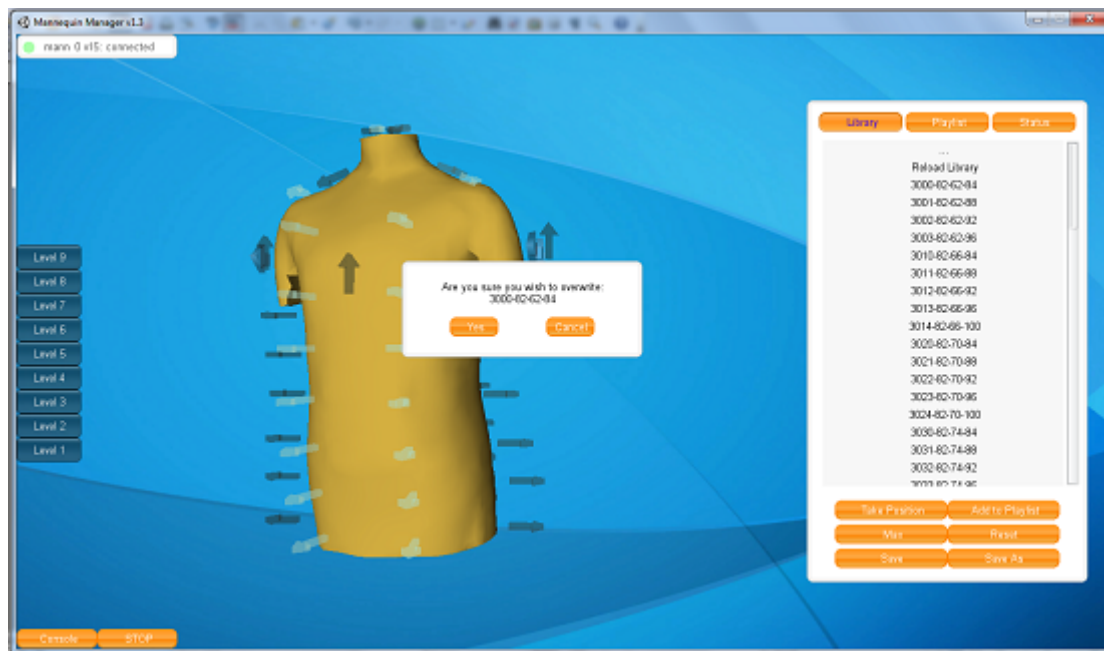
## Lisad

### Lisa 1. Mannequin Manageri ekraanipildid

Käesolevas lisas on toodud ekraanipildid erinevatest Mannequin Manageri tööolukordadest, mis ei ole töö suure mahukuse tõttu kajastatud põhiosas.

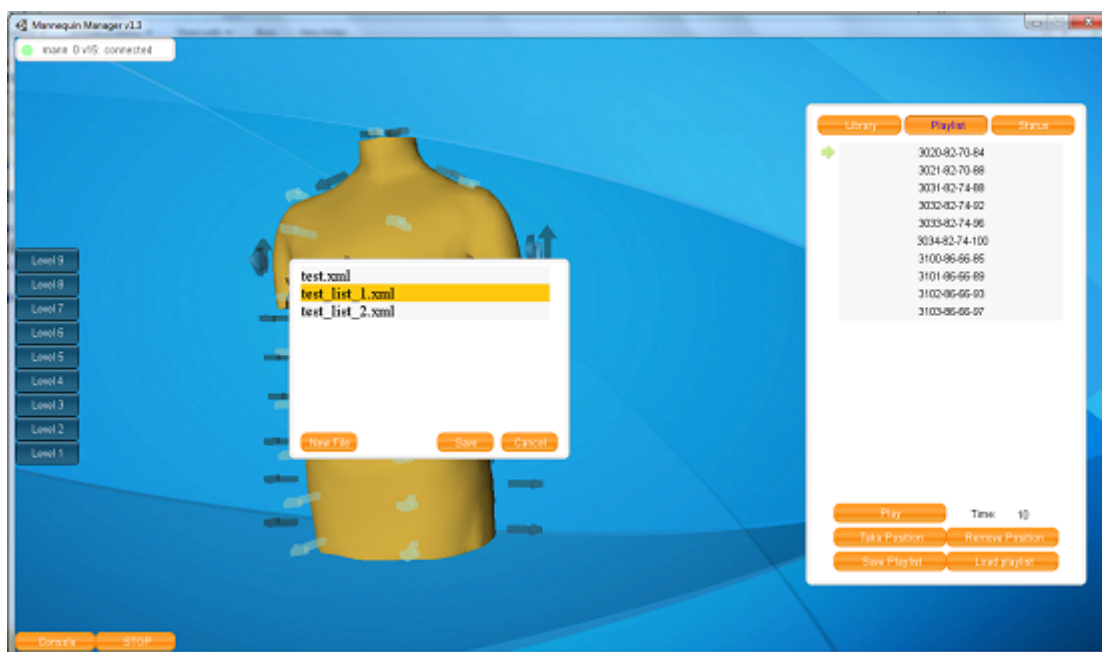


Joonis 14. Uue asendi salvestamine.

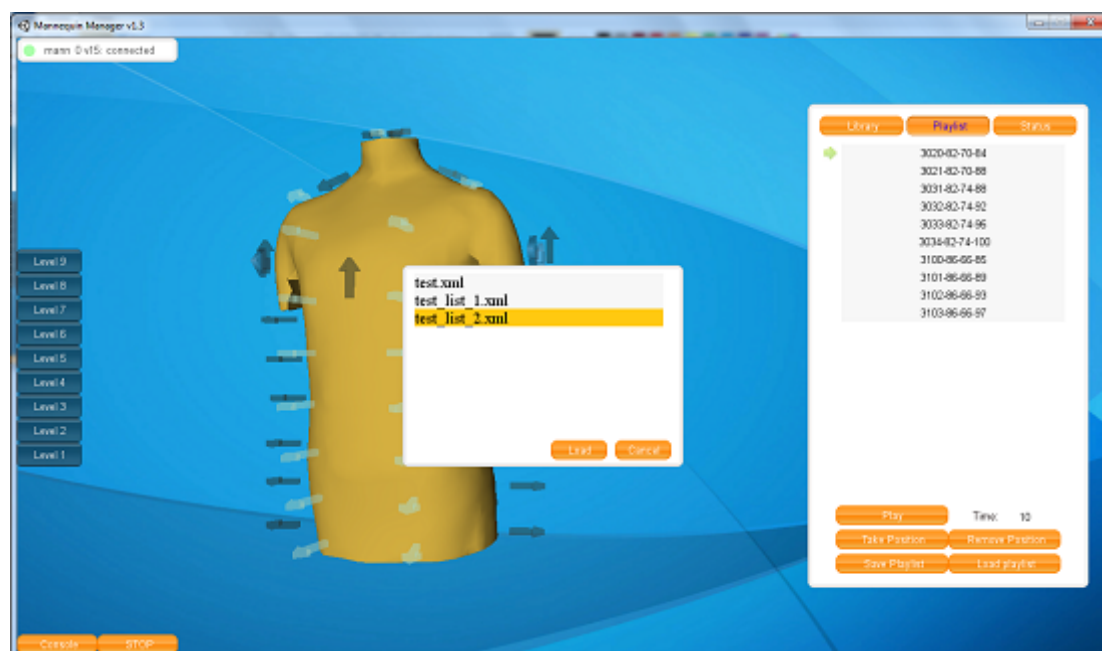


Joonis 15. Vana asendi üle kirjutamine.

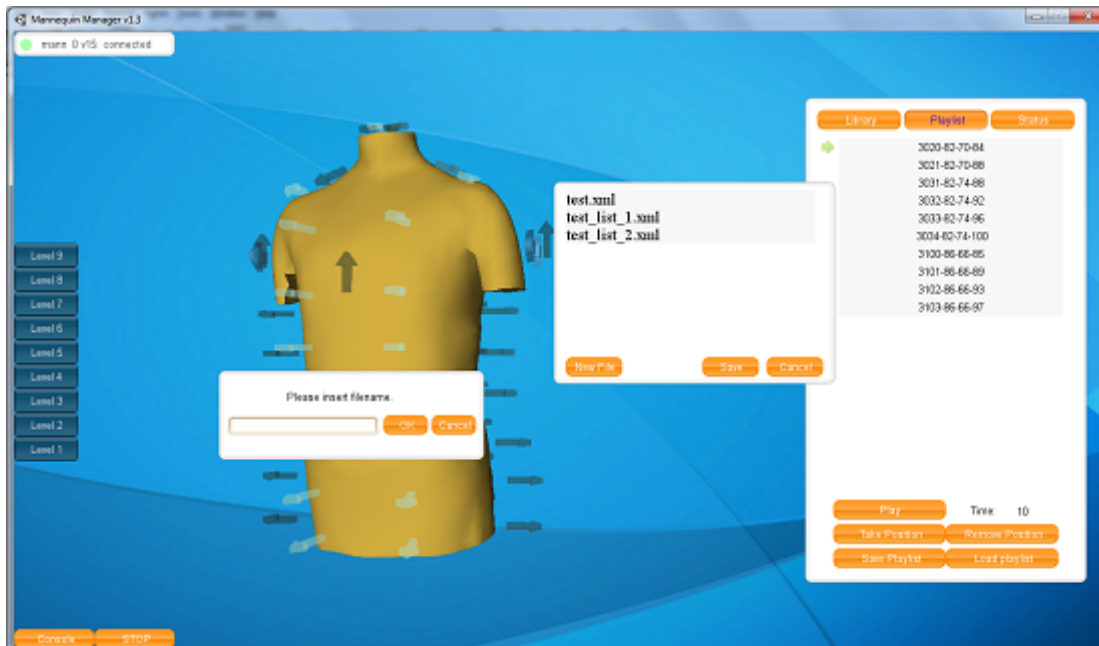




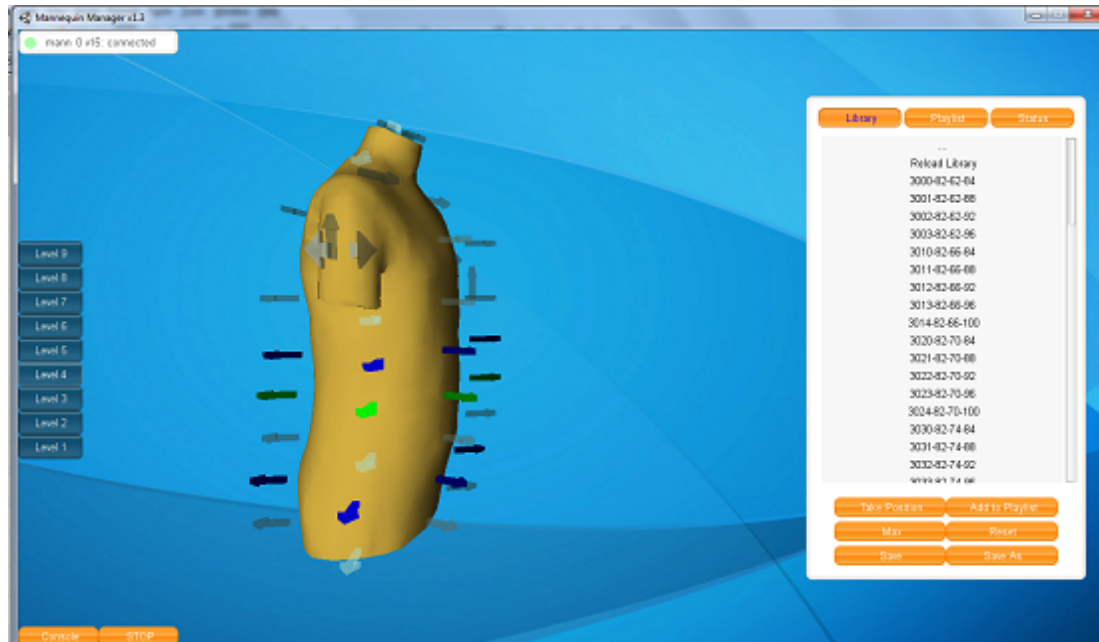
Joonis 16. Esitusloendi salvestamine.



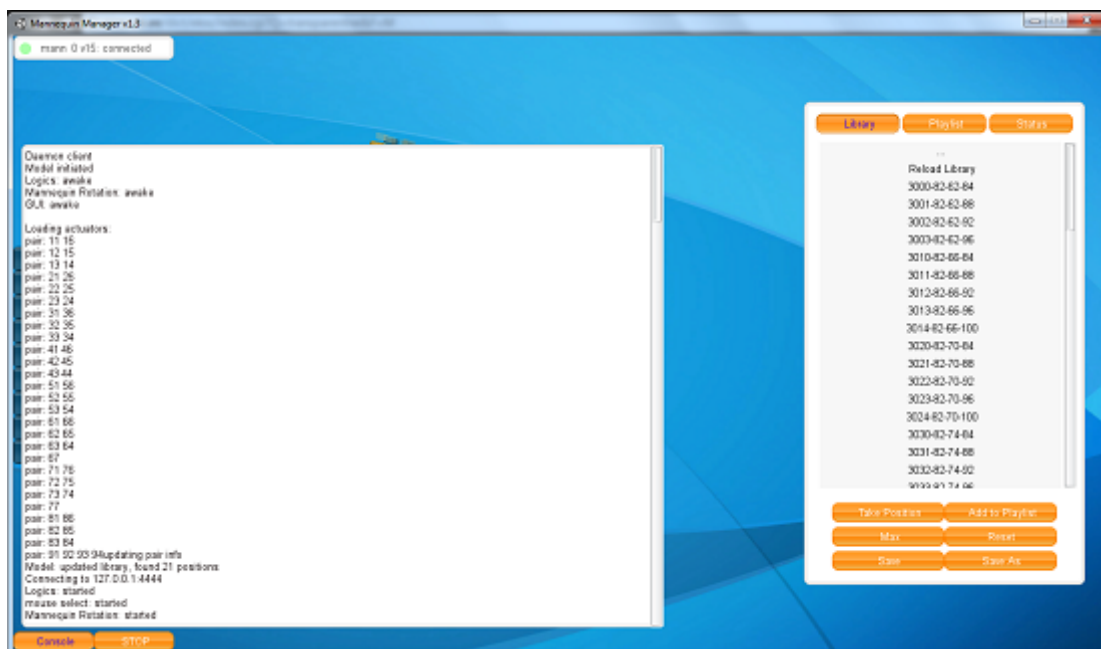
Joonis 17. Esitusloendi laadimine.



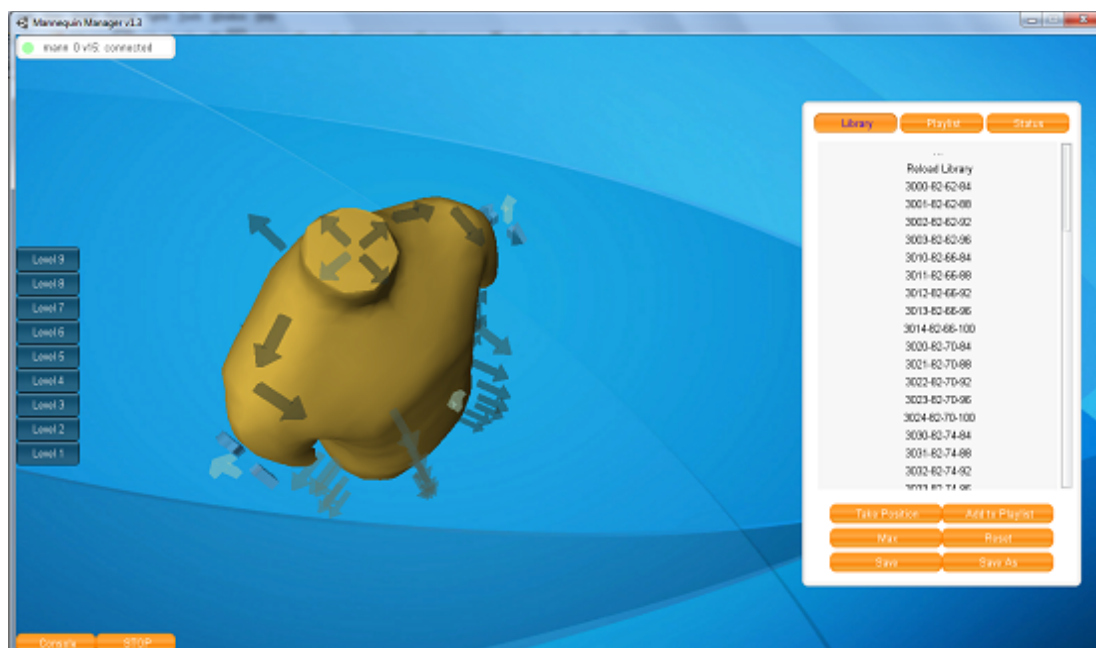
Joonis 18. Kasutaja saab dialoogaknaid nihutada.



Joonis 19. Ekraani vasakust servast *Level* nuppe kasutades valitakse korraga terve tasandi mannekeeni ümbritsevad nooled. Ctrl klahv võimaldab järjest mitme tasandi valimist.



Joonis 20. Konsool võimaldab näha teateid Mannequin Manageri tööst.



Joonis 21. Paremat hiireklahvi all hoides võib kasutaja 3D mudelit pöörata soovitud nurga alla.